

EXHIBIT 10

INTERNET ARCHIVE

WayBackMachine

5 captures

14 Dec 07 - 24 Jan 10

http://weblogs.java.net/blog/terrencebarr/archive/2007/11/sooo_what_about_1.html

Go

APR JUL

24

2007 2008

java.net

The Source for Java Technology Collaboration

User:

Password:

Login

Register

Login help

My pages

Projects

Communities

java.net

Get Involved

java-net Project

Request a Project

Project Help Wanted Ads

Publicize your Project

Submit Content

Get Informed

About java.net

Articles

Blogs

Events

Also In Java Today

java.net Online Books

java.net Archives

Get Connected

java.net Forums

Wiki and Javapedia

People, Partners, and Jobs

Java User Groups

RSS Feeds

Search

Web and Projects:

»

Online Books:

»

Advanced Search

Terrence Barr's Blog

«Developer Days News-Flash: James Gosling, and over 60 papers submitted | Main | Update: Great content coming for Java Mobile & Embedded Developer Days »

sooo, what about Google Android and phoneME?

Posted by terrencebarr on November 06, 2007 at 02:54 PM | Comments (13)

- Update 1: Dalobor Topic has just posted [this](#) about the Android license.
- Update 2: Check out [this](#) MacWorld article in which Rich Miner talks about the Android non-fragmentation pledge. Proprietary extensions and consistency in the platform? Sounds like wishful thinking to me.

Over the last 24 hours very little news and lots of opinion and speculation hit the airwaves about Google's Android platform and the [Open Handset Alliance](#). Our very own community member Sean Sheedy has [this](#) very detailed blog entry to offer.

After digesting the first barrage of information and talking to a couple of folks in the industry I thought I'd offer up my own personal thoughts about Android and the OHA.

First, at a high level, I think the Android announcement shows that the era of proprietary and closed mobile platforms and networks is finally drawing to an end (remember that Google has also [indicated](#) they will participate in the auction of the 700 Mhz spectrum). And that's goodness because locked platforms and networks controlled tightly by vendors have been a huge barrier to innovation and ignored the needs of consumers for too long (["Ten things I want from my phone"](#)). I commend Google for taking this step.

But once you dig a little deeper into what Android is and what the business model and motivations are I think you'll be left scratching your head a bit. The Android [press release](#) and OHA website offers surprisingly little information (and I was a bit annoyed by this [video](#) which offers meaningless babble and a dog sitting in for Andy Rubin).

The consensus seems to be that Android is a Linux- and Java-based stack including the low-level OS, middleware, application layers, and some basic applications - at first glance very similar to Sun's [JavaFX Mobile](#) (but without a key component [FX Script](#)). It appears to be highly customizable allowing, as Eric Schmidt puts it, "thousands of different phone models" (in contrast to JavaFX Mobile which places a strong emphasis on consistency and predictability). The stack is licensed under Apache v2 and an early access SDK will be available next week.

OHA founding members are Google, T-Mobile, HTC, Qualcomm, Motorola, and Aplan. Other notables on the current member list are Esmerlec, Intel, KDDI, LG, NTT DoCoMo, Samsung, Sprint Nextel, and Telefonica. That's impressive but also noteworthy for who is not on that list: Nokia, SonyEricsson, and most carriers around the world (except for Japan where OHA has good coverage with DoCoMo and KDDI). Also, remember that being part of a press release is easy and says very little about how much skin you will actually have in the game at a later point in time.

So what does it all mean?

Right away, the questions that come to mind are: How is this different from the existing 40+ Linux-based handset platforms? The [Limo Foundation](#), the [LIPS Forum](#), and other efforts in this space? And who is the customer for this? Here a very interesting blog by Carl Howe ["What's the Point of Google's Phone?"](#).

But every more perplexing is the question of what the business case and motivation is for OEMs and carriers to be part of Google's push. In the mobile space there is a very strong established pattern of maintaining tight control of the platform itself, the content on the platform, and the data streams to and from the platform to support business models that drive up the ARPU (average revenue per user). OEMs and carriers rely on proprietary features, walled gardens (directly, but more frequently indirectly), and business arrangements to implement these models. That's not to say they are not investigating new business models but history suggests that the closed model is something they are very comfortable with.

Google, on the other hand, built its business around the open Internet and freely accessible content and applications running on standardized platforms (such as the web browser). Of course, the ad business accounts for something like 95% of Google's revenue so I assume Google's main motivation for Android is to build a new ad delivery platform with a potential for billions of new users and a reach they just never could have achieved on

UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA
TRIAL EXHIBIT 7459.1

CASE NO. 10-03561 WHA
DATE ENTERED _____
BY _____
DEPUTY CLERK

the desktop.

Fair enough, but Google's model is diametrically opposed to the approach of OEMs and carriers. If Google's goal is to drive mobile users to the Google portal and use Google services anywhere at any time on any platform where does that leave the carrier and OEM? They basically become a commodity platform and a dumb data pipe with the purpose of driving Google eyeballs. Sure, Google can offer them revenue sharing and other incentives but fundamentally they would lose the leverage and monetization points they have built a huge business around.

This is not a trivial problem. In fact, it is one of the key reasons the mobile space today is what it is, with a multitude of vendor-specific platforms, software stacks, proprietary features, and limitations. It's also a situation Sun and the Java ME ecosystem have been having to deal with since the release of KVM in 1997. One of the key reasons Java ME has been extremely successful (admittedly not without its share of pain and frustration) is because Sun's position allowed it to be a neutral 3rd party that provided a platform and ecosystem but didn't need to enforce any particular business model. So it was a natural symbiosis for OEMs and carrier.

And this point precisely is where the Android news leaves a big void. It's like fitting a square peg into a round hole. I don't see how Google's business model and the OEM and carrier model align sufficiently to make this a happy partnership.

And what's ominous for developers is that the Android license choice and [FAQ](#) explicitly encourages proprietary extensions on the platform as well as the removal of functionality - which spells trouble for delivering a widespread integrated platform that allows developers to build compelling services easily and quickly. This seems to have "fragmentation" of the worst degree written all over it. Imaging thousands of unique Android-based platforms ... Java ME fragmentation, with all its quirks, may be a piece of cake compared to that.

Which leaves us with the final question. What about Sun and Java ME?

Obviously, Sun isn't part of the OHA at this point - I am disappointed at this but I assume the OHA and Sun had disagreements over business and strategic issues that made a joining of Sun at that time impossible. That might change but I am not privy to those discussions.

Next, why is Java ME not mentioned in the press release? Well, since Aplix and Esmertec are part of OHA I think that is a strong indication Java ME is a key component of the Android platform. And Google being famously secretive I am not surprised they are not volunteering that information. Fact is, Google recognized Java to be a necessary part of success. It literally guarantees Java remains the dominant platform for years to come.

So what about [phoneME](#)? phoneME is fully open and licensed under GPLv2 in the [Mobile & Embedded Community](#). Sun believes the GPL license is the best choice for reasons of compatibility with GNU/Linux as well as to ensure ongoing innovation benefits everyone in the ecosystem while maintaining a consistent platform. Google has chosen Apache v2 for Android which means the licenses are incompatible and therefore phoneME cannot be part of Android at this point in time. If and how this might be resolved in the future is speculation and not a particularly useful exercise right now.

But with Sun's vision of the [JavaFX Mobile](#) as the next-generation mobile software platform Java wins in any case

Welcome to exciting times.

-- Terrence

PS: I'll take the opportunity to remind you of the upcoming [Java Mobile & Embedded Developer Days](#) conference. Registration is now open. Check it out [here](#).

Powered by Qumana

Bookmark blog post:  [del.icio.us](#)  [Digg](#)  [dz](#)  [DZone](#)  [Furl](#)  [Reddit](#)

Comments

Comments are listed in date ascending order (oldest first) | [Post Comment](#)

- It depends on what actually the "full stack" is, that Google is advertising. It can be deduced by now that it includes the Linux kernel (GPL), GNU libc or uclibc (LGPL), and WebKit (LGPL). None of those components can be easily turned into proprietary software, as Google would like people to believe. So I would not take the ALv2-only-stack talk very seriously, as it makes as much sense as the dog video.

From the site, it seems that Google wants something like Maemo+real phone. Given the kind of apps Google makes money on, they'd probably be more interested in a fast ECMAScript execution environment on such devices, than a Java bytecode one, and the ALv2 licensed GWT 'stack' includes Java-to-ECMAScript converters, afaik. So, I'd expect the Java language to play a role, but not necessarily the JVM, except as an optional deployment module.

It is mildly amusing to watch Google emulate Sun in the early ME days, with all the good intentions, cheering for proprietary software built on open source/standards, and all that. From a developer perspective, Apple has Google beat in this round, as they already have their semi-proprietary devices on the market, with a dev kit coming soon, while Google has a dev kit coming soon, with the class of probably subtly incompatible semi-proprietary devices about a year away.

I, for one, can't wait for Google to announce their entry into the gaming console market with an alliance between Atari, Commodore, and Disney. :)

Posted by: robilad on November 07, 2007 at 03:12 AM

- Hey Dalibor,

Thanks for pointing that out. Yes, the whole notion of "open" is very blurry in the Android announcement and probably for good reason. Keeping an opening for the different agendas and goals of the participants is a challenging undertaking, to say the least.

Posted by: terrencebarr on November 07, 2007 at 03:26 AM

- Perhaps there are not that many carriers on the list because Google is planning to become a global carrier themselves? Their business model might even allow them to give away (hardware and) connectivity for free, provided the whole world uses "their" gateway to the internet.

Posted by: marrs on November 08, 2007 at 05:18 AM

- Possible. But becoming a carrier with significant market share is very hard. The 700 Mhz spectrum is only free in some U.S. markets. Pretty much everywhere else in the world Google would have to partner. Remember that carriers have worked for years to get into the position they have now where they have an oligopoly to airwaves. New technologies might be an option to break that lock (wimax) but carriers are already jousting for dominance in those as well.

Posted by: tarrencebarr on November 08, 2007 at 05:24 AM

- Agreed, I don't claim to be knowledgeable in this area and you're right about this being very difficult. It's a marketplace with many initiatives, where big players are battling for years now. Looking at the future of computing, mobile connected platforms will become more and more important. Time will tell!

Posted by: marrs on November 08, 2007 at 07:08 AM

- robilad,

You clearly don't understand the requirements of these OSS licenses... Or you're just trying to spread FUD. The modified GPLv2 license that the Linux kernel uses already allows for proprietary drivers (kernel modules). For example, both nVidia and ATI have proprietary drivers for Linux, yet they aren't required to release their code under the GPL.

GNU's libc is developed under the GPL with a "linking" exception. How are Adobe, IBM, Sun (formerly with Java), HP, BEA and others capable of running closed source apps on Linux that link with glibc? Reason — you are not bound by the GPL when you link with glibc.

As you've said WebKit is LGPL. This means that anyone can use WebKit in their app. The only thing the LGPL requires you to do is redistribute any changes you make directly to the WebKit source code. 99.9999% of mobile phone app developers are not interested in modifying the WebKit code base in a proprietary way — they want to USE that component, but not directly modify its source. BTW — you do know that the iPhone/Safari uses WebKit right? How in the world will anyone create proprietary apps for iPhone!?!?!?

Lastly, the code that Google themselves are going to release is licensed under an Apache-style license. This is code that a proprietary app developer may directly extend. In this case, there is not need to open source your code because of the Apache license.

This entire "full stack" can indeed be EASILY used to create proprietary apps. You're highly misinformed.

Also, I disagree that the JVM won't play a big role on the gphone. ECMAScript will be prevalent too, but any heavy-lifting apps (or apps with significant animation) will require the power of Java.

-Bryan

Posted by: prime21 on November 08, 2007 at 11:37 AM

- Prime21,

I think you latched onto the wrong thing here. What Dalibor said is that a number of parts of the Android stack can't be "turned" into proprietary software — Google doesn't have the right to change the licenses. Of course you can link proprietary software to Linux, libc, and Webkit ... but that doesn't make it a proprietary stack and Google seems to imply. That's all. By the way, Dalibor is a well-known open source expert and a very thoughtful guy so you shouldn't be accusing him in that manner.

Cheers,

— Terrence

Posted by: terrencebarr on November 08, 2007 at 01:12 PM

- Bryan, please read the Google Android FAQs carefully. Or see my linked light hearted blog piece. And please take it easy, OK?

The point is not that the Android stack can be used to create proprietary apps, the point is that Google claims that the stack itself is a turnkey solution for proprietary, locked down phones, should the phone

vendor want to differentiate itself.

That's not quite correct, and that's where Google's Android marketing folks maybe should have spent a couple of minutes more talking with the developers, rather than making funny dog videos.

For an example of a (phone) company that got bitten by acting on the idea that you can treat the Linux kernel as a turnkey component for proprietary phone software stacks, without actually complying with the GPL on it, I'd like to point you to the recent court case in Germany that Skype lost.

Posted by: robble on November 08, 2007 at 02:04 PM

- Okay. Sorry. Re-reading your original post makes sense now :-).

Posted by: prime21 on November 08, 2007 at 02:11 PM

- No worries, happens to me all the time in the heat of the moment, too. ;)

Posted by: robble on November 08, 2007 at 02:13 PM

- In Googlespeak "open" means "open to submit wholesale to Google" or alternatively "open to spying by Google", nothing else.

Posted by: jwenting on November 08, 2007 at 10:49 PM

- Google Android announcement, means to me that it is just in time for Java SE to embrace the mobile platforms !

Step one, opensource JavaFX Mobile platform as well (under GPL with classpath exception, say) to provide an android stack complement .

Step two, let's start an "embrace and extend campaign" by bringing Java SE stack on the "upcoming major highend mobile OS" namely : Windows Mobile, Android and maybe ... even iPhone (depending on the upcoming SDK capabilities).

Doing so, you would simply bring WORA to mobile devices, because AFAIK, today Windows Mobile applications (even if using cf.net) are locked to Windows Mobile and tomorrow Android application will also be to some extent (because of specific API used) lock to Android platform, and so should be iPhone applications !

Let's get Java back to its root : highend mobile devices ! As a start, all we need is an OpenJDK "portage" tutorial, something that clearly describe step by step all that has to be brought to OpenJDK code to introduce a new platform (module by module, file by file, etc). The one that can contribute such a tutorial will change the future of Java ! Anyone with skills here ?

Posted by: bjb on November 14, 2007 at 07:59 AM

- I'm pretty much in line with Terrance's posting.

All that matters at the end of the day is the carriers and their business model. You can have all the open platforms you want, but they are going to do what they want. Hopefully Google's marketing muscle will be able to change things and create a more innovative and scalable development model for mobile apps. But I'll believe it when I see the carriers allowing it.

Steve Haney
Tira Wireless

Posted by: smhaney on November 20, 2007 at 01:47 PM

This work is licensed under a [Creative Commons License](#).



Powered by
Movable Type 3.01D

[java.net RSS Feeds](#)



[Feedback](#) | [FAQ](#) | [Press](#) | [Terms of Use](#)
[Privacy](#) | [Trademarks](#) | [Site Map](#)

Your use of this web site or any of its content or software indicates your agreement to be bound by these [Terms of Participation](#).

Copyright © 1995-2008 Sun Microsystems, Inc.

O'REILLY COLLABNET
Powered by Sun Microsystems, Inc.,
O'Reilly and CollabNet